TÉCNICO LISBOA

AUDIOVISUAL COMMUNICATION

Laboratory Session: Audio Processing and Coding

The objective of this lab session is to get the students familiar with audio processing and coding, notably psychoacoustic analysis which is exploited in MP3 coding. For that purpose, a Matlab Graphical User Interface (GUI) is described here[1] which demonstrates some of the principles behind perceptual audio coding.

1. SOME BASIC PSYCHOACOUSTIC CONCEPTS

MPEG-1 Audio Layer III, commonly referred to as MP3, is a digital audio coding standard, performing lossy audio coding. MP3 was designed by the Moving Picture Experts Group (MPEG) as part of its MPEG-1 Audio standard and later extended in the MPEG-2 Audio standard, notably to consider more audio channels and lower sampling frequencies.

The main principle behind MP3 lossy coding is perceptual coding. This means that psychoacoustic principles and models are exploited not only to identify the frequency components which are not audible at all due to full masking but also to determine how much the precision of the audible frequency components may be reduced due to the consideration of the human auditory system characteristics; both actions target the elimination of information which is deemed “perceptually irrelevant” and thus the consequent rate savings. Irrelevant information is identified by the encoder using a psychoacoustic model, which considers phenomena such as the absolute hearing thresholds as well as frequency and temporal masking.

The human auditory system, see Figure 1 left), is not equally sensitive to the full audible audio spectrum, meaning that different frequency components are subject to different perceptual sensitivity. Thus, the power of a sound wave is not the only factor influencing its perceived loudness, and it is, in fact, possible for a sound wave to appear to a human louder than other with a larger power. Figure 1 right) shows the so-called *equal loudness curves* which represent what the power of a sound must be in order to be perceived as the same loudness of a 1 kHz tone at a certain power level. For example, the same loudness is perceived for 10 dB at 1 kHz and 30 dB at 100 Hz, as humans are less sensitive at 100 Hz than at 1 kHz.

The *absolute hearing threshold* or *hearing threshold in quiet* is characterized as the minimum power that a given frequency tone must have to be perceived in a noiseless environment (this means when that tone happens alone), see the dashed line in Fig. 1.

[1] This application has been developed by André Guarda, in November 2017, in the context of a PhD course.

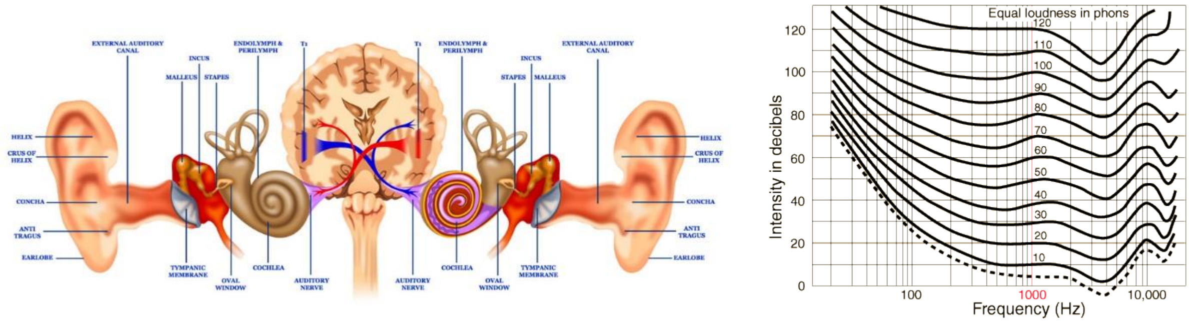


Figure 1: left) Human auditory system; right) Equal loudness curves[2].

Frequency masking (also known as *simultaneous masking*) refers to the phenomenon where one sound becomes inaudible or audible with less precision because of the presence of one or more other simultaneous sounds. Fig. 2 left) shows an example of frequency masking for a single tone and the corresponding masking threshold. Without the masking tone, the other sound would be audible as its power level is above its absolute hearing threshold. However, when both sounds are played simultaneously, the sound at 250 Hz masks the other sounds that fall beneath the corresponding masking threshold. If the tone would be above the masking threshold, it would be audible but it could be coded with less precision, e.g. more quantization, as the masking threshold was raised regarding the absolute hearing threshold and thus reduced precision is needed due to the reduced sensibility.

In addition, masking may also occur when the sounds are not temporally simultaneous but happen close to each other. In effect, just before and just after a masking sound, there is a brief period where other sounds may be perceptually inaudible or have a reduced audibility. This phenomenon is referred as *temporal masking* and is represented in Fig. 2 right).

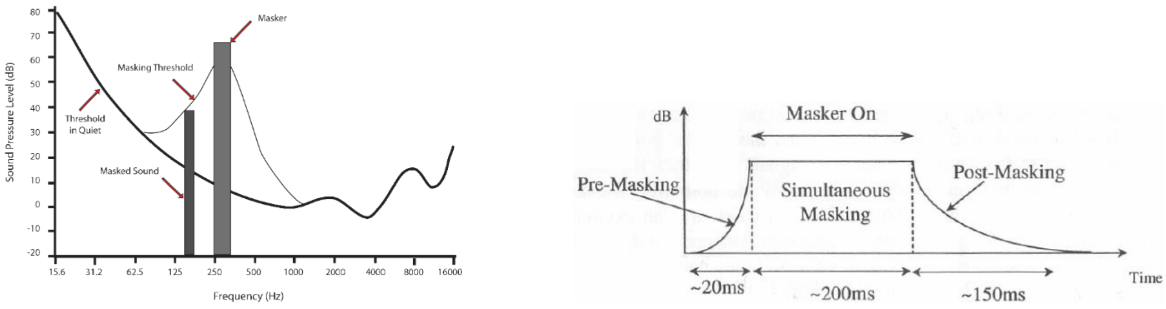


Figure 2: left) Frequency masking[3]*; and right) Temporal masking*[4].

2. THE GRAPHICAL USER INTERFACE

The developed Matlab Graphical User Interface (GUI) is illustrated in Fig. 3 to Fig. 7. With the use of this GUI, the user may experience by listening some basic psychoacoustic phenomena that are exploited in audio perceptual encoder, as well as the compression capabilities of an MP3 encoder. Tabs may be changed by pressing the desired button in **B1**. To reset the GUI, press button **B2**.

[2] https://guitartheory.wordpress.com/2006/11/06/the-sound/

[3] https://en.wikipedia.org/wiki/Psychoacoustics

[4] Y. Lin, and W.H. Abdulla, "Technical Report: Audio Watermarking for Copyrights Protection," Technical Report, School of Engineering, The University of Auckland, 2007.

2.1. Part 1: Auditory Sensitivity

The GUI opens with the first tab already selected, as shown in Fig. 3. In this first part, the user can experience that the human auditory sensitivity is different for different frequencies.

Initially, to calibrate the audio volume of the computer, press button **B3** to listen to the calibration tone, while simultaneously adjusting the computer audio volume until the tone is audible.

The user can listen to the so-called *power staircase signal*, which plays a selected frequency tone consecutively in 10 steps of -5 dB each (this means successively lower power tones), starting at 70 dB, by pressing button **B4**. To select the frequency of these power staircase tones, use the radio buttons **R1**. In panel **P1**, the power and frequency of the signal currently playing is shown. To stop the sound, simply press button **B5**.

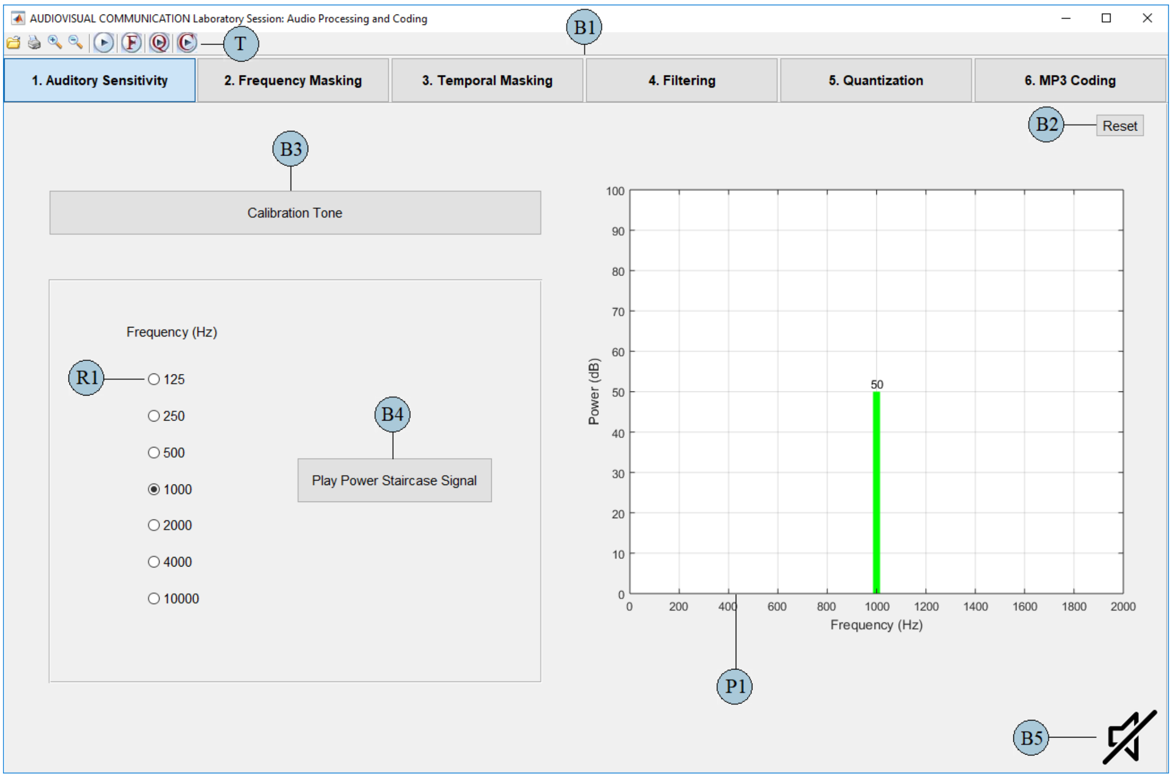


Figure 3: Snapshot of the Auditory Sensitivity GUI.

2.2. Part 2: Frequency Masking

In this part, the existence of audio signals (maskers) that mask the auditory perception of other simultaneous audio signals may be experienced. A snapshot of the Frequency Masking GUI is shown in Fig. 4.

The user can define a masking tone signal by choosing its frequency (between 20 and 20000 Hz) and power (between 0 and 100 dB) in the input boxes **I1** and **I2**, respectively. Then, to listen to this masking signal, press button **B3**. Then, the masked signal is defined as a power staircase signal, similarly to the previous section, with a frequency defined in the input box **I3** and an initial power which is the same as the masking signal. Button **B4** plays the defined masked signal.

Finally, the user can listen to both audio signals together by pressing button **B5**. Panel **P1** shows the frequency and power of the signal(s) currently playing. To stop the sound, simply press button **B6**.

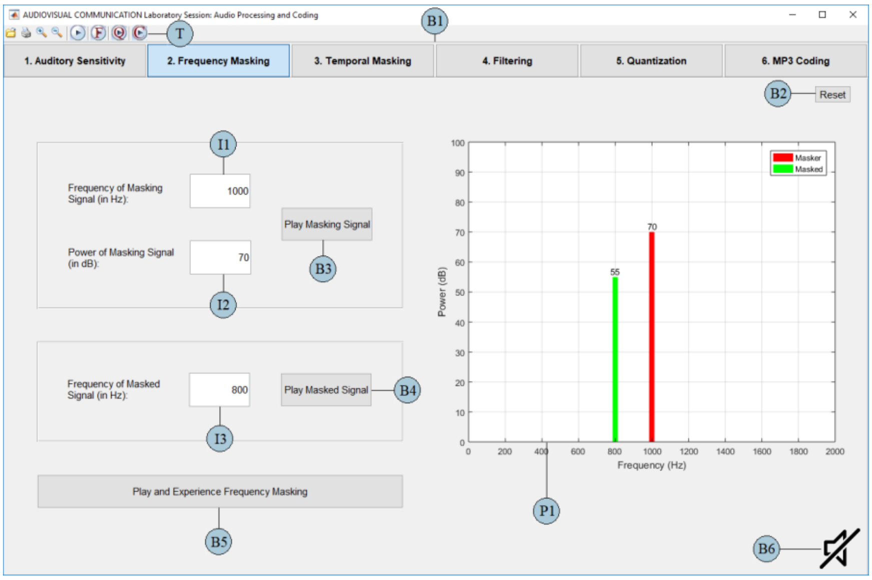


Figure 4: Snapshot of the Frequency Masking GUI.

2.3. Part 3: Temporal Masking

In addition to frequency (simultaneous) masking, the user can also experience the temporal masking phenomenon; a snapshot of the Temporal Masking GUI is shown in Fig. 5.

To play this demonstration, press button **B3**. Here, noise bursts of 250 ms are played intertwined with single tones at 2000 Hz playing for a duration of 10 ms, starting with a power 80 dB and successively reducing in 10 steps of -4 dB each, similarly to the power staircase signal used in the previous section. First, the noise burst that plays here the role of a temporal masking signal is played alone; after, the noise burst is followed by a tone signal, with a time gap between the noise and the tone signal of 100 ms, 20 ms or 0 ms; this time gap value may be chosen with radio button **R1**. In panel **P1,** the signal and noise bursts are plotted in time. To stop the sound, simply press button **B4**.

The temporal masking effect increases when the time gap between the masker and masked signals is reduced and when the masked signal power decreases.

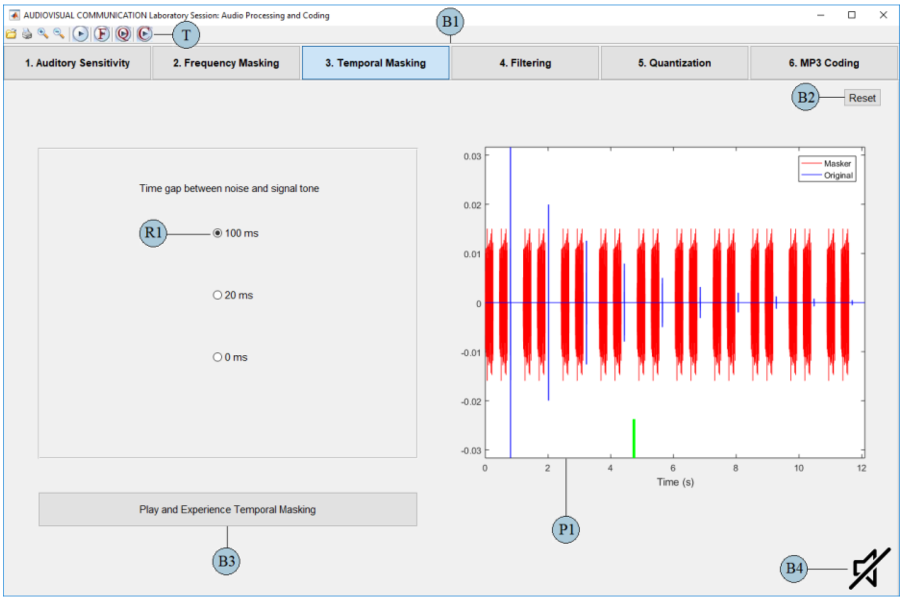


Figure 5: Snapshot of the Temporal Masking GUI.

2.4. Part 4: Filtering

This part shows the effect of filtering an audio signal. A snapshot of the Filtering GUI is presented in Fig. 6.

The user can select an audio file using the toolbar (**T**) button 📂 or button **B3**. The audio signal over time is plotted in blue in panel **P1** while the absolute hearing threshold is plotted in green. By clicking button **B4**, the panel alternates between displaying the audio signal over time and its spectrum, as shown in Fig. 6. If the selected audio signal is stereo, only the left channel is presented. Furthermore, the user can listen to the audio signal by pressing the appropriate button in the set **B6** or the toolbar (**T**) button ▶.

The user can select the type of filter to apply to the audio signal, notably Low-pass or High-pass, with radio button **R1**, and the desired cutoff frequency in the input box **I1**. Then, the user has to press button **B5** to apply the selected filter. The filtered audio signal is plotted in panel **P2**, and the difference between the original and filtered signals (the 'error' signal) is plotted in panel **P3**. The user can listen to the filtered audio signal by pressing **B7** or the toolbar (**T**) button F, as well as to the error signal pressing **B8**.

The user can specify the starting time for playing the audio signal by clicking in the respective panel, **P1**, **P2** or **P3**. Additional options are available, namely pausing, resuming and stopping the audio signal by using the appropriate **B6**, **B7** or **B8** buttons.

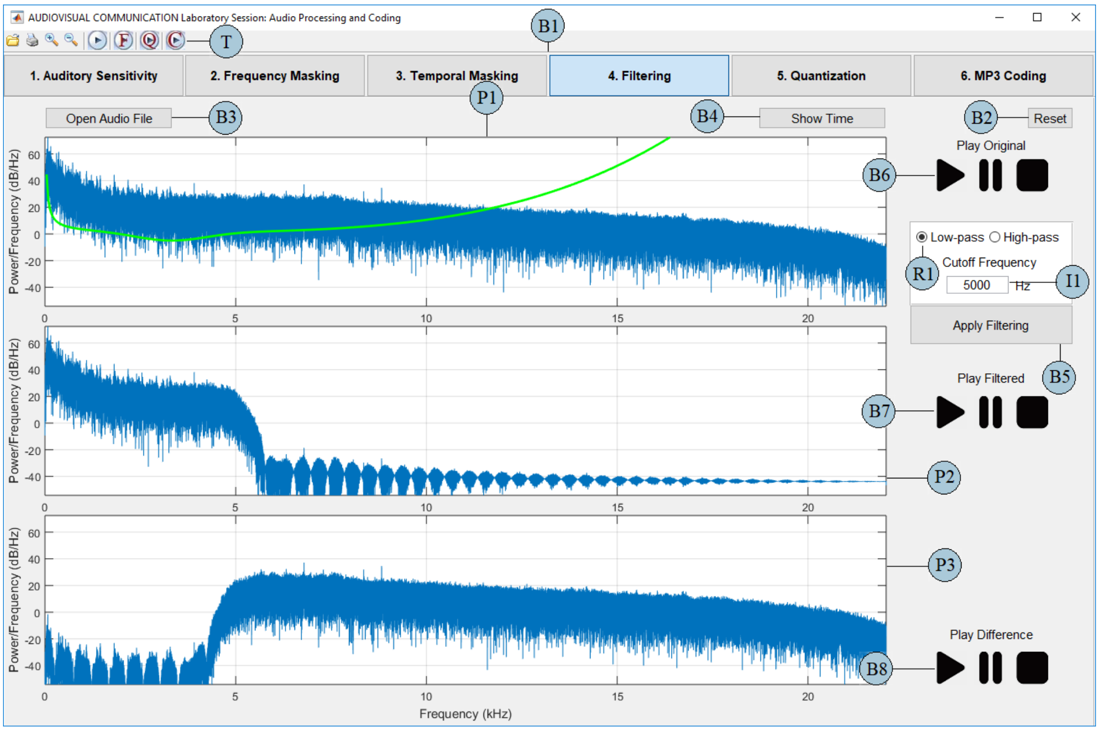


Figure 6: Snapshot of the Filtering GUI.

2.5. Part 4: Quantization

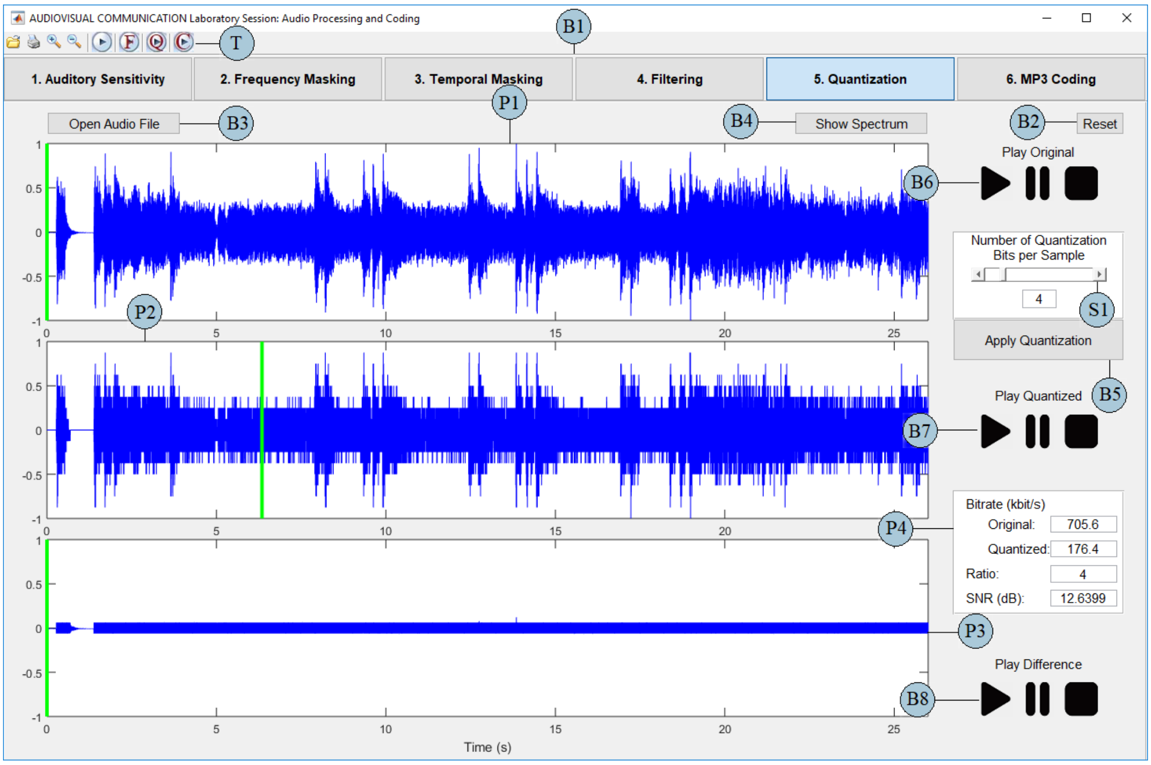
This part allows to experience the effects of quantizing an audio signal. Fig. 7 shows a snapshot of the Quantization GUI.

Similarly to the previous part, the user can select an audio file using the toolbar (**T**) button or button **B3**. The audio signal is plotted in panel **P1** and the user can listen to the audio signal by pressing button **B6** or the toolbar (**T**) button. By clicking button **B4**, the panel alternates between the signal over time and its spectrum. The user can select the number of quantization bits per sample in slider **S1**. Then, the quantization is performed by pressing button **B5**. The obtained bitrate, compression ratio and Signal to Noise Ratio (SNR) are given in panel **P4**, the quantized audio signal is plotted in panel **P2**, and the difference between the original and quantized signals (the quantization error) is plotted in panel **P3**. The user can listen to the quantized signal by pressing button **B7** or the toolbar (**T**) button, as well as to the error signal by pressing **B8**. The additional audio playing controls are also available.

2.6. Part 5: MP3 Coding

This part helps the user to experience the coding capabilities of an MP3 audio encoder; its GUI is shown in Fig. 7.

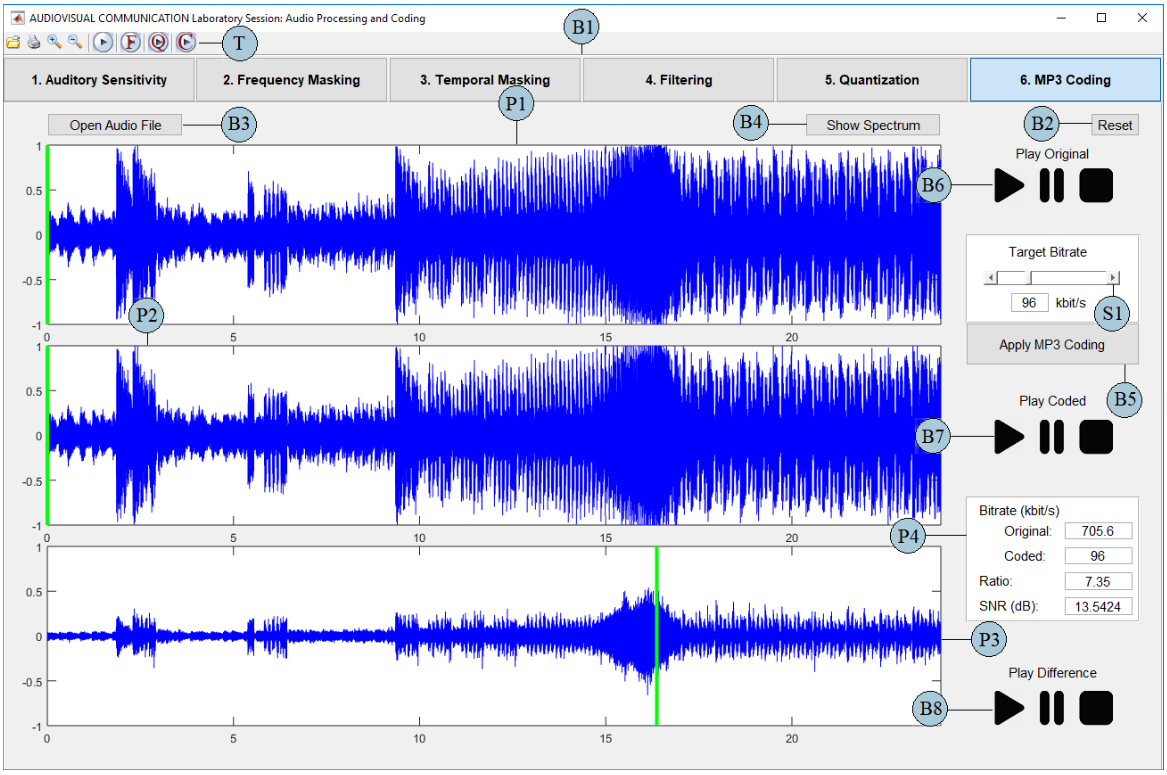
The user can select an audio file using the toolbar (**T**) button or button **B3**. The audio signal is plotted in panel **P1** and the user can listen to the audio signal by pressing button **B6** or the toolbar (**T**) button. By clicking button **B4**, the panel alternates between displaying the signal over time and its spectrum. In slider **S1**, the MP3 coding rate is specified. However, this target bitrate might slightly differ from the finally obtained bitrate due to the accuracy of the used rate control algorithm. To code the audio signal, the user has to press button **B5**. The obtained bitrate, compression ratio and SNR are given in panel **P4**, the compressed signal is plotted in panel **P2**, and the difference between the original and decoded signals (the coding error) is plotted in panel **P3**. The user can listen to the decoded signal by pressing **B7** or the toolbar (**T**) button, as well as the error signal by pressing **B8**. The additional audio playing controls are also available.



Number of Quantization Bits per Sample: 4

Bitrate (kbit/s)
Original: 705.6
Quantized: 176.4
Ratio: 4
SNR (dB): 12.6399

Figure 7: Snapshot of the Quantization GUI.



AUDIOVISUAL COMMUNICATION Laboratory Session: Audio Processing and Coding

Target Bitrate: 96 kbit/s

Bitrate (kbit/s)
Original: 705.6
Coded: 96
Ratio: 7.35
SNR (dB): 13.5424

Figure 8: Snapshot of the Coding GUI.

AUDIOVISUAL COMMUNICATION

Laboratory Session: Audio Processing and Coding

Date ___________ **Day of the week** ____

Nº ___________ **Name** ____

Nº ___________ **Name** ____

Try addressing the following questions in a speedy way but do not rush if you do not fully understand what you are doing/experiencing; in that case, ask for help. The most important is to learn …

Open the "Audio Processing and Coding Lab" Graphical User Interface (in Matlab, double click the folder "CAV Laboratory Session - Audio Processing and Coding", then open the file "Run_Program.m" and press "Run"). Start by adjusting the audio volume in your computer to 50%, then click the button "Calibration Tone" to make sure the sound is audible.

1. Auditory Sensitivity: For each available frequency, press the "Play Power Staircase Signal" button and count the number of power steps that you can hear and register the data in the table below. The "Power Staircase Signal" button plays consecutive tones of a specific frequency in 10 steps of -5 dB each, starting at 70 dB. What do you observe? Relate this observation with the absolute hearing threshold, as seen in Fig. 1 right).

Frequency (Hz)	**125**	**250**	**500**	**1000**	**2000**	**4000**	**10000**
Number of Audible Power Steps							

____

____

____

____

____

____

____

____

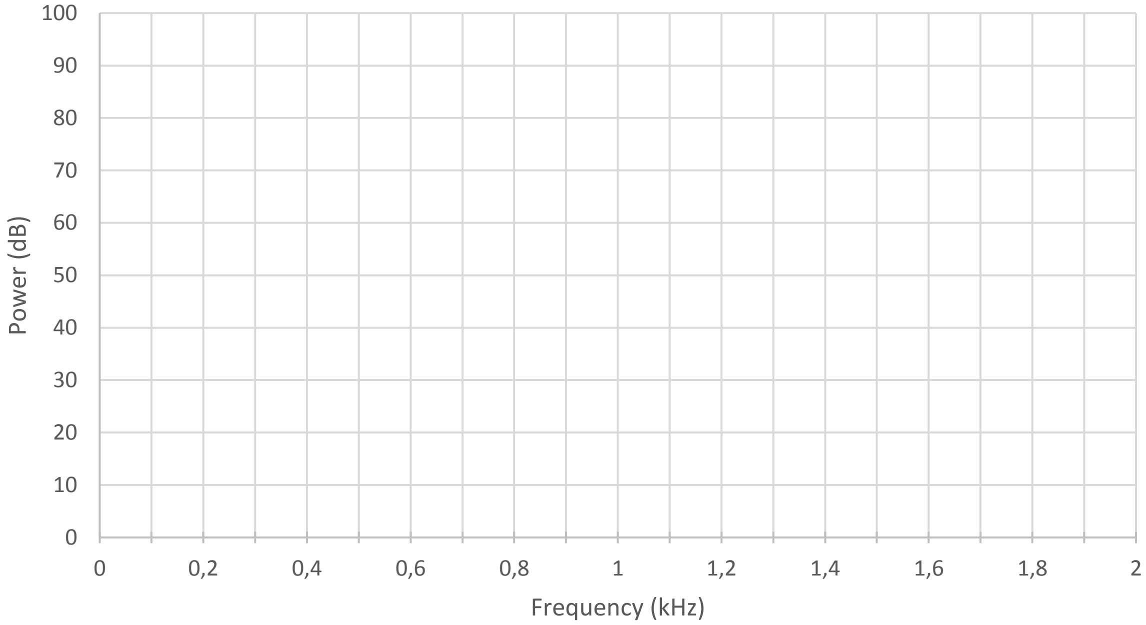
____

____

____

____

2. Frequency Masking: Set the frequency of the masking signal to 1000 Hz and the power to 70 dB. Your objective is to draw the masking curve below which other sounds are not perceptible. To do so, choose successive frequencies for the masked signal above and below the masking signal frequency, e.g. 950, 900, 800, 700, 600, 500, 400 and 1050, 1100, 1150, 1200, 1300, 1500, 1800 Hz. The masked signal will take successive powers in a staircase sequence, starting at the same power of the masking signal. Then, by listening to the mixture of both signals and observing the plot in the panel, determine, for each frequency, the power value for which the masked signal is fully masked (this means, when you stop hearing it). Register all values and draw the masking curve. Comment on the results and explain how this study can be useful for audio compression.



3. Temporal Masking: A temporal masking example is available to show the masking effects of noise played right before a tone signal. This signal is a power staircase signal at 2000 Hz, starting at power 80 dB, and reducing the power in 10 steps of -4 dB each, with each step having a duration of 10 ms. Before each step, 2 noise bursts 250 ms long are played; the first one is played alone for comparison, and then another noise burst is played followed by the tone signal. The goal is to perceive if the tone signal is audible or is masked when varying the time gap between the noise and the tone signal. Play and experience the temporal masking for each of the time gaps available (100 ms, 20 ms or 0 ms), which may be chosen with the radio buttons R1, and record the number of audible steps in each case. Comment on your observations as the time gap decreases.

Time gap (ms)	**100**	**20**	**0**
Number of Audible Steps			

____

____

____

____

____

____

____

____

____

For the remaining questions, decrease the audio volume in your computer to a comfortable level when listening to the audio samples.

4. Filtering: Select one of the audio files from the folder “Low band”. These audio files are characterized by having a rather low bandwidth, this means they essentially miss high frequencies. Visualize the signal spectrum to confirm this feature, and estimate yourself a cut-off frequency (this means a frequency for which the low-pass filtered signal should perceptually sound similar to the original signal) by comparing the signal spectrum with the absolute hearing threshold (in green). Starting with this estimate, experiment with different cut-off frequencies for the low-pass filter until the filtered signal sounds perceptually essentially similar to the original signal and compare with your estimation. Recall that you may use the complementary high-pass filter to see (on the panel) and hear (on the headphones) the filtered information. Now select other files from the folder “High band” and repeat the experience. These audio files are richer in frequency. This should allow you to experience what type of ‘sounds’ correspond to each frequency region.

Which of these two types of audio signals can be coded with higher efficiency (lower bitrate) with MP3? Comment on your observations and explain how a perceptual audio codec, such as MP3, can take advantage of this cut-off frequency information, notably to use a lower sampling frequency to reduce the coding rate.

5. Quantization: Select a "Low band" audio file. Vary the number of quantization bits per sample (Slider S1), quantize the signal and listen to it. Write down a subjective quality score for the quantized sound using a 1-5 scoring range where 1 is bad and 5 is indistinguishable from the original sound (which uses 16 bits per sample). Repeat for a "High band" audio file.

Comment on the subjective quality of the audio, notably annoying artifacts, for the successive numbers of quantization bits per sample, relating to the characteristics of the audio signal. Is this type of quantization (this means uniform for all frequencies) a good way to achieve compression? Why? How do the MPEG-1 Audio codecs address this problem?

Low band signal: Number of quantization bits per sample	**Low band signal: Subjective score (1-5)**	**High band signal: Number of quantization bits per sample**	**High band signal: Subjective score (1-5)**
4		**4**	
6		**6**	
7		**7**	
8		**8**	
9		**9**	
10		**10**	
12		**12**	

6. MP3 Coding, Quality versus bitrate: Select a “Low band” audio file. Vary the target bitrate (Slider S1), code the signal with MP3 and listen to it. Write down a subjective quality score for the coded sound using a 1-5 scoring range where 1 is bad and 5 is indistinguishable from the original sound (which uses 16 bits per sample). Repeat for a “High band” audio file.

Comment on the subjective quality of the audio, notably annoying artifacts, for the several bitrates, relating to the bandwidth characteristics of the audio signal.

Low band signal: Bitrate (kbps)	**Low band signal: Subjective score (1-5)**	**High band signal: Bitrate (kbps)**	**High band signal: Subjective score (1-5)**
32		**32**	
48		**48**	
64		**64**	
96		**96**	
128		**128**	
192		**192**	

7. MP3 Coding, Quantization versus MP3 Coding: For multiple audio files, quantize them, and also code them with MP3, using the same bitrate for both operations. Listen to both the processed audio and compare the quantized and MP3 decoded signals for similar bitrates to comment on the quality and artifacts, notably for lower bitrates. Relate your observations with the way MP3 perceptual coding works, notably in terms of adaptive quantization.